Electronic Version

Stylesheet Version v1.1.1

Description

ASYNCHRONOUS INTERFACE METHODS AND APPARATUS

BACKGROUND OF INVENTION

[0001] FIELD OF THE INVENTION

[0002] The present invention relates to asynchronous interface design, and more particularly to improved asynchronous interface methods and apparatus.

[0003] BACKGROUND OF THE INVENTION

[0004] An asynchronous interface may be employed to exchange signals between multiple clock domains. In such instances, it is often difficult to verify that the asynchronous interface functions properly across various alignments of the multiple clocks running at different frequencies. In many cases, such verification does not occur. For example, when signals traveling across an interface experience different delays (e.g., due to differing logic, wiring or other delays), a signal "launched" across the interface first

may actually be latched at a receiving side of the interface by a clock cycle that occurs after a clock cycle employed to latch a subsequently transmitted signal. Behavioral simulations, gate-level simulations and static timing tools all may have difficulty modeling such conditions. Accordingly, conventional modeling typically cannot predict whether a state machine on a receiving side of an asynchronous interface will function under "real world" conditions that may produce signal transmission failures.

[0005] A need therefore exists for asynchronous interface methods and apparatus that can simulate the actual failing conditions of an interface, and/or that allow such failing conditions to be mitigated and/or eliminated.

SUMMARY OF INVENTION

[0006] In a first aspect of the invention, a first method is provided for aligning signals from a first receiver located in a first clock domain to a second receiver located in a second clock domain. The first method includes the steps of creating a programmable delay element between the first and second receivers, and selectively adding delay via the programmable delay element to the signals until the signals are aligned.

[0007] In a second aspect of the invention, a second method is

provided for aligning signals transmitted between a first receiver located in a first clock domain and a second receiver located in a second clock domain. The second method includes the steps of (a) providing at least one selectable delay for each of a first plurality of signal lines adapted to receive signals transmitted from the first receiver to the second receiver; and (b) providing at least one selectable delay for each of a second plurality of signal lines adapted to receive signals transmitted from the second receiver to the first receiver. The second method further includes the steps of (c) testing operation of the first and second receivers in response to differing delays between signals transmitted between the first and second receivers; (d) determining one or more delays that cause the first and second receivers to exchange signals without errors; and (e) employing the one or more delays during subsequent transmission of signals between the first and second receivers.

[0008] Numerous other systems, methods, computer program products and data structures also are provided. Each computer program product described herein may be carried by a medium readable by a computer (e.g., a carrier wave signal, a floppy disc, a compact disc, a DVD, a hard

drive, a random access memory, etc.).

[0009] Other features and aspects of the present invention will become more fully apparent from the following detailed description, the appended claims and the accompanying drawings.

Brief Description of Drawings

[0010] FIG. 1 is a schematic diagram of a conventional asynchronous interface.

[0011] FIG. 2 illustrates a first exemplary asynchronous interface system provided in accordance with the present invention.

[0012] FIG. 3 illustrates a second exemplary asynchronous interface system provided in accordance with the present invention.

[0013] FIG. 4 is an exemplary embodiment of the first programmable delay stage of FIG. 3.

Detailed Description

[0014] FIG. 1 is a schematic diagram of a conventional asynchronous interface 100 for transmitting signals between a first receiver such as a state machine 102 (that operates in a first clock domain 102a) and a second receiver such as a state machine 104 (that operates in a second clock domain 104a) via a plurality of signal lines 106a-d. The first

and second signal lines 106a-b may be employed to transmit signals from the second state machine 104 to the first state machine 102. Such signals may be latched into the first state machine 102 via a first plurality of metastability or similar latches 108a, 110a and 108b, 110b, respectively. Likewise, the third and fourth signal lines 106c-d may be employed to transmit signals from the first state machine 102 to the second state machine 104. Such signals may be latched into the second state machine 104 via a second plurality of metastability or similar latches 108c, 110c and 108d, 110d, respectively.

[0015] During a typical simulation of the operation of the asynchronous interface 100 of FIG. 1, each signal line 106a-d is assumed to have the same delay (e.g., 0) associated therewith. Accordingly, two or more signals transmitted across the asynchronous interface 100 simultaneously are assumed to arrive at a latch 110a-d at the same time. In practice, the signal lines 106a-d may each have differing logic, wiring or other delays; and the relative position in time between two signals at a receiver side of the asynchronous interface 100 may not be the same as the relative position in time between the two signals when the signals were launched from a transmitter side of the asyn-

chronous interface 100. For example, by the time the latches 108a-b, 110a-b (1) capture the logic state of two signals transmitted from the second state machine 104 to the first state machine 102; and (2) settle to a state, the two signals may be received by the first state machine 102 an entire clock cycle apart (even though the signals were transmitted from the second state machine 104 at approximately the same time). Similarly, by the time the latches 108c-d, 110c-d (1) capture the logic state of two signals transmitted from the first state machine 102 to the second state machine 104; and (2) settle to a state, the two signals may be received by the second state machine 104 an entire clock cycle apart (even though the signals were transmitted from the first state machine 102 at approximately the same time).

[0016] To accommodate the differing logic, wiring or other delays of the signal lines 106a-d, the first and second state machines 102, 104 should be resilient enough to function despite an unintended shift in time of as much as a receiver clock cycle between signals transmitted across the asynchronous interface 100. However, for the reasons stated above, conventional simulation techniques typically cannot simulate such potential failure conditions. Further,

conventional asynchronous interfaces generally do not allow unintended shifts between signals to be corrected or compensated for without redesign.

[0017] FIG. 2 illustrates a first exemplary asynchronous interface system 200 provided in accordance with the present invention. As shown in FIG. 2, the asynchronous interface system 200 includes the components 102-110d of the asynchronous interface 100 of FIG. 1, as well as a supplemental asynchronous interface device (SAID) 202 coupled between the first and second state machines 102, 104.

[0018] With reference to FIG. 2, the SAID 202 includes a first clock domain portion 204 and a second clock domain portion 206. The first clock domain portion 204 includes (1) a first delay circuit 208 coupled to the first signal line 106a for selectively introducing a first clock domain clock cycle delay to a signal traveling on the first signal line 106a; and (2) a second delay circuit 210 coupled to the second signal line 106b for selectively introducing a first clock domain clock cycle delay to a signal traveling on the second signal line 106b. That is, each delay circuit 208, 210 may introduce no delay or a one clock cycle delay (in the first clock domain 102a) to a signal traveling on the signal line 106a, 106b.

[0019] The second clock domain portion 206 includes (1) a third delay circuit 212 coupled to the third signal line 106c for selectively introducing a second clock domain clock cycle delay to a signal traveling on the third signal line 106c; and (2) a fourth delay circuit 214 coupled to the fourth signal line 106d for selectively introducing a second clock domain clock cycle delay to a signal traveling on the fourth signal line 106d. That is, each delay circuit 212, 214 may introduce no delay or a one clock cycle delay (in the second clock domain 104a) to a signal traveling on the signal line 106c, 106d.

[0020] Each delay circuit 208-214 may comprise, for example, a multiplexer-type circuit for selecting between a delayed and non-delayed signal path. For example, the first delay circuit 208 includes a multiplexer or similar logic 216 adapted to select between a non-delay signal path 218 and a signal path that includes a latch 220. The latch 220 is adapted to be latched via a receiver clock (e.g., a clock of the first clock domain 102a in the case of the first delay circuit 208). Multiplexer selection signals may be provided in hardware, software or a combination thereof; and other delay signal configurations may be employed (e.g., additional delay paths, paths with longer delays, etc.).

[0021] In operation, the SAID 202 may be employed to delay any (or all) signals that travel across an asynchronous interface (e.g., via signal lines 106a-d) by one receiving clock cycle. For example, simulating interface behavior with many randomly generated tests having signals transmitted across the interface system 200 with different combinations of delays may eventually detect problems with the asynchronous interface system 200 and/or the state machines 102, 104. Alternatively, all possible combinations of random signal delays may be employed during simulated signal transmission across the interface system 200 to exhaustively verify operation of the asynchronous interface system 200 and/or state machines 102, 104. In this manner, the robustness of each state machine 102, 104 to unintended signals delays of up to a clock cycle may be verified (e.g., by skewing signal arrival times under some or all possible conditions). Problems with an asynchronous interface thereby may be found with a logic simulator as part of a normal verification process. Further, such verification may be performed at the behavior-level or gate level.

[0022] In one or more embodiments of the invention, the SAID 202 need only be employed during simulation to allow

more complete testing/simulation of an asynchronous interface during signal exchange between state machines. Such an approach is simpler than attempting to simulate the various delays associated with each signal line of the asynchronous interface; and may be implemented as one or more computer program products. Each computer program product may include, for example, a netlist of components of the SAID 202 or another simulatable representation of the SAID 202.

[0023] To provide further verification options, the SAID 202 may be implemented in hardware as part of an asynchronous interface. For example, the SAID 202 may be included within an asynchronous interface as shown in FIG. 2, and employed to run tests through use of a hardware emulation platform (e.g., by randomly or systematically introducing delays to signals traveling between the state machines 102, 104 so to determine the response/resiliency of the state machines 102, 104). Additional testing also may be performed on any production hardware.

[0024] When implemented in hardware, the SAID 202 also may be employed to repair the operation of an otherwise defective interface. For example, the SAID 202 may be employed to permanently introduce a clock cycle delay to one or more

signal lines of an asynchronous interface to correct defective operation of the interface. Such delay selection may be achieved for example, by blowing one or more fuses or anti-fuses (not shown) provided with the SAID 202 or via any other suitable technique.

[0025] While the present invention has been described with reference to an asynchronous interface that employs four signal lines 106a-d, it will be understood that the present invention may be employed to selectively add delays to any number of signal lines of an asynchronous interface. In additional to adding one clock cycle of delay to one or more signal lines, the present invention may be employed to add additional delays to one or more signal lines (e.g., multiple clock cycle delays).

[0026] FIG. 3 illustrates a second exemplary asynchronous interface system 300 provided in accordance with the present invention. The second exemplary asynchronous interface system 300 is similar to the first exemplary asynchronous interface system 200 of FIG. 2, but employs N signal lines 106a-n between the first and second state machines (not shown in FIG. 3) operating in clock domains 102a, 104a, respectively. In general, any number of signal lines may be employed.

[0027] Further, the second exemplary asynchronous interface system 300 includes a plurality of programmable delays for each of the signal lines 106a-n provided via a supplemental asynchronous interface device (SAID) 302. In the embodiment shown, each signal line 106a-n includes up to N-1 programmable delay stages 304a-n. That is, a signal traveling on a signal line 106a-n may be delayed by any amount between 0 and N-1 receiver clock cycles (e.g., in the clock domain of the state machine receiving the signal). Each delay stage may comprise, for example, a plurality of latches that are clocked via a receiver clock, and/or similar hardware to that described with reference to the delay circuits 208-214 of FIG. 2.

[0028] FIG. 4 is an exemplary embodiment of the first programmable delay stage 304a of FIG. 3. The programmable delay stages 304b-n may be similarly configured. As shown in FIG. 4, the programmable delay stage 304a includes a multiplexer or similar logic 402 adapted to select between a non-delay signal path 404a and signal paths 406a-m that include increasing numbers of latches 408 (e.g., from one latch for signal path 406a to n-1 latches for signal path 406m). Each latch 408 is adapted to be latched via a receiver clock (e.g., a clock of the first clock

domain 102a in the case of the first programmable delay stage 304a). Multiplexer selection signals may be provided in hardware, software or a combination thereof; and other delay signal configurations may be employed.

[0029] When implemented in hardware (e.g., when not employed solely for simulation purposes) the selection of the number of delays per signal line may be performed, for example, by blowing one or more fuses, anti-fuses, or the like. Providing up to N-1 selectable delays per signal line allows greater flexibility for altering signal line delays to correct interface errors, and in most cases, the sequence of N interface signals may be altered to correct errors without requiring redesign. Likewise, the provision of up to N-1 selectable delays during simulation allow further testing options.

[0030] In one embodiment of the invention, every control vector of the NxN-1 delay latch matrix associated with the SAID 302 may be employed to test operation of the asynchronous interface system 300 exhaustively. For example, if the asynchronous interface system 300 employs five signal lines, a control vector of (0 0 0 0 0) would cause no delay on any signal line, a control vector of (0 0 0 0 1) would cause one receiver clock cycle delay on the 5^{th} sig-

nal line, a control vector of (0 0 0 1 2) would cause a one receiver clock delay on the 4th signal line and a two receiver clock delay on the 5th signal line, etc. One possible algorithm for performing such an exhaustive test may be implemented as follows:

[0031]

```
Control_vector = - 1
Do {
     Increment Control_vector
     Run test
}
```

[0032] Other testing algorithms similarly may be employed.

[0033] Testing in accordance with the present invention may be performed in any manner. For example, testing may be performed on a standalone tester under tester control. Likewise, a control vector (e.g., Control_vector) may be visible to software in an actual use environment. Accordingly, if a problem is detected during testing, a diagnostic routine may be employed to identify/employ the control_vector required to correct the problem (e.g., the system 300 may be self-healing and/or employ self-healing software/hardware). For example, the inventive system may be designed to automatically test for, detect and/or repair interface problems in an actual use environment

(e.g., by employing control vectors to test the interface and detect the error, by identifying a control vector that will correct the error and by employing the control vector to select the appropriate number of delays per signal line to correct the error). Such a process may be implemented, for example, as one or more computer program products (e.g., operable with one or more controllers and/or processes of a testing system (not shown)).

[0034] The foregoing description discloses only exemplary embodiments of the invention. Modifications of the above disclosed apparatus and method which fall within the scope of the invention will be readily apparent to those of ordinary skill in the art. For instance, other numbers of signal lines and/or delays per signal lines may be employed.

[0035] Accordingly, while the present invention has been disclosed in connection with exemplary embodiments thereof, it should be understood that other embodiments may fall within the spirit and scope of the invention, as defined by the following claims.